Amendments to the Specification:

On page 1 before line 1, add the following:

CROSS-REFERENCES TO RELATED APPLICATIONS

This application is related to and claims priority from Japanese Application No. 2000-371486, filed December 1, 2000 and Japanese Application No. 2000-381652, filed December 11, 2000.

Replace the paragraph beginning on page 1, line 17, with the following amended paragraph:

A DVD (digital versatile disc) is a medium which ~~is~~ has about 7 times ~~larger than~~the capacity of a CD (compact disc)~~ in capacity~~. As a medium that records data on a DVD, there are DVD-RAM, DVD-R, and DVD-RW. Recordable media are ~~fields most expectable from now on~~becoming increasingly more popular, as seen in the recent growth of CD-R.

Replace the paragraph beginning on page 2, line 1, with the following amended paragraph:

First, an overview of demodulation processing of ~~data recorded~~recorded data on DVD-RAM will be shown by using Fig. 3. The main data 310 that has been recorded is divided ~~per~~ into blocks of 2048 bytes. ~~In each, a~~As shown in Fig. 4, each block includes 4 bytes of identification data (ID) 401, 2 bytes of IED (ID Error Detection Code) 402 that is an ID error detecting code parity, and 6 bytes of RSV (reservation bytes) 403 that is a reservation area, in addition to the main data (2048 bytes) for a data stream of 2060 bytes ~~are added~~. Furthermore, 4 bytes of error detecting code parity (EDC) 404 is added to the end of a data stream of 2060 bytes~~ of this data stream, then~~. The resulting 2064 bytes constitute ~~of~~ data unit 1 (before scramble) ~~1 (~~304 and is organized into twelve rows of 172 bytes per row as shown in Fig. 4~~) is configured~~. Next, according to a scramble rule determined with the value of the ID part of the data unit 1, scramble processing is performed ~~to~~ on the 2048 bytes of the main data part 310 to produce~~, and it becomes~~ a data unit 1 (after scramble) ~~1 (~~305~~) which is 12 rows x 172 bytes~~. Sixteen instances of these data units (after scramble) 1 are combined to form a data field comprising 192 rows x 172 columns bytes, e.g., (see data field 306 in Fig. 5. Next, 16 rows of Parity of Outer-code (PO) data are added (step 306) to the 192-row data field to produce a 208-row x 172 column data field. Next, ten bytes of Parity of Inner-code (PI) is added (step 307) to each of the 208 rows, extending each row to 182 bytes, to form one ECC block. Then, a PO interleave operation to interleave the sixteen rows of PO data (182 bytes wide) with the original 192-row data field. More specifically, a PO row is added at the end of each instance of data unit 1 (i.e., twelve rows) in the 192-row data field to form sixteen 13-row

data units 2 (step 308). After an 8/16 modulation operation is performed to convert the 8-bit data to 16-bit data, twenty-six SYNC codes are added, per one data unit 2 to form sixteen data units 3 (step 309).~~Furthermore, 16 bytes of Parity of Outer code (PO) is added to each of 172 columns (lengthwise) of 192 rows x 172 bytes that are configured by 16 data units 1 by making 16 blocks of these data units (after scramble) 1 (305) mutually combined. Next, 10 bytes of Parity of Inner code (PI) is added to each of 208 rows (widthwise) containing PO to form one ECC block (307) which is a 182 rows x 208 bytes cross Reed Solomon code. Then, PO interleave which puts each of 16 rows of PO parts (including 16 rows of PI for them) into data units 1 as a 13th row is performed to form 16 data units 2 (308) (that is, one data unit 2 consists of 13 rows). After 8/16 modulation that converts all data into 16 bits from 8 bits is given thereafter, 26 SYNC codes are added per one data unit 2 to form 16 data units 3 (309).~~

Replace 4 paragraphs beginning on page 3, line 6, with the following amended paragraphs:

First, 2064 bytes of data ~~is~~are configured ~~by,~~as shown in Fig. 4, comprising 2048 bytes of main data (310), 4 bytes of identification data (ID401), IED (ID error Detection Code) 402 that is an ID error detecting code parity, 6 bytes of RSV (reservation bytes) 403 that is a reservation area, and 4 bytes of EDC (error Detection Code) 404. A data unit (after scramble) 1 (305) is a data unit ~~by making~~in which the 2064 bytes of data are organized as twelve rows of ~~into the form of~~ 172 bytes each ~~x 12 rows~~, where the ~~and scrambling~~ 2048 bytes of main data part are scrambled.

~~16~~Sixteen data units (after scramble) 1 (306) ~~is~~constitute a 192 rows of 172 bytes each ~~x 192 rows~~ in a data field 306 and are configured by combining ~~16~~sixteen instances of data units 1 (305) as shown in Fig. 5.

An ~~1~~ECC block 307 is a unit of error correction processing in DVD and is a product code (known as a cross Reed Solomon code). The product code ~~will be~~is formed as follows. As shown in Fig. 6, an outer-code is configured by defining ~~16~~sixteen data units 1 as information data and ~~making~~ treating each of 172 columns ~~be~~ as RS (208, 192, 17); thus ~~is formed (~~16 ~~bytes~~ rows of parity of outer-code PO_502 ~~is~~ are added as error correction data~~)~~. After that, an inner-code is formed by ~~making~~ treating the resulting 208 rows ~~including PO502 be~~ as RS (182, 172, 11) in the same direction as the direction of a main data list and the recording direction in a DVD (parity of inner-code PI_501 is added as error correction data). A data unit 2 (308) is a data unit after the formation of the ECC block 307, and is a 13-row x 182 byte~~s x 13 rows~~ data unit that is configured by ~~inserting~~ interleaving each of the sixteen~~16~~ rows of PO_502 ~~into each data unit including PI501 (PO interleave)~~among the data units as shown in Fig. 7.

A data unit 3 (309) is a data train obtained by adding eight kinds (SY0-SY7) of synchronizing signals 801 (SYNC) at the head of each block at every 91 bytes of the data unit 2 as shown in Fig. 8. ~~, f~~For example, SY0 (SYNC code 0) in the head of the unit, SY1-SY4 cyclically inserted in the head of each row, and SY5, SY6 and SY7 inserted for identification of each row, and performing 8/16 modulation converting 8-bit data into 16-bit data. Moreover, it is possible to identify a position of data, which is reproduced, in

the data unit 3 with the generating pattern of SYNC 801 from an addition method of this SYNC 801 at the time of reproduction.

Replace 2 paragraphs beginning on page 4, line 25, with the following amended paragraphs:

Moreover, the list of main ~~data recorded~~recorded data on the DVD ~~is never changed~~does not change during the data format conversion process such as scramble, ECC encoding, and PO interleave. Hence, the list and sequence of ~~data recorded~~recorded data on the DVD are equal to the list and sequence of main data.

Therefore, the ~~data recorded~~recorded data on DVD is reproduced in a reverse process of data processing performed in Fig. 3.

Replace 3 paragraphs beginning on page 5, line 9, with the following amended paragraphs:

The 8/16 demodulation is performed while determining a position in the data unit 3 (902) by using SYNC 801, and the data unit 2 (903) is generated. Next, the PO interleave is ~~released~~ reversed (deinterleaved) while determining a position of the data unit 2 in an ECC block by using ID 401, and an ECC block 901 corresponding to the ECC block 307 before error correction is formed.

The reproduction data from the DVD ~~has an error to occur by~~is susceptible to error from various factors. Hence, error correction to a maximum of 5 bytes per inner-code (normally, although each data constituting the code is expressed by "word" or "symbol" in error correction, here, the data is explained by using "byte" as an example of a data unit) is performed in the error correction to inner-code having the same list as a data list recorded in the DVD. The correction of an error, included in each outer-code, to a maximum of 16 bytes is performed, and error is removed by outer-code error correction. After that, error correction data PI 501 and PO 502 are removed from the ECC block 901, 16 data units 1 (305) shown in Fig. 5 is obtained.

The data of the data unit 1 (305) ~~releases the scramble~~de-scrambled (304) and is restored into ID, IED, RSV and 2048 bytes of main data (303) again after performing the error detection processing by using EDC so as to verify that erroneous correction is not performed in the error correction processing.

Replace 4 paragraphs beginning on page 6, line 10, with the following amended paragraphs:

In the error correction processing in the reproduction of a DVD that is described by using Fig. 9, the error correction of a PI having an identical list with the one of the ~~data recorded~~recorded data in the DVD is performed.

The relationship of data placement on DVD and data placement on ECC block is effective in view of making it possible to correct a comparatively long burst error by performing ~~eraser~~ error correction, which uses a result as error position information in outer-code error correction after not letting errors, continuously arising as shown by black parts in Fig. 10C, dispersed, and several uncorrectable error arising in the inner-

code error correction. The vanishing correction method is an error correction method of performing the correction by determining outer-code error positions ~~by~~ using the correction result of the inner-code as described in this specification.

Nevertheless, an uncorrectable error arises in plenty of outer-codes when short burst errors randomly arising as shown by black parts in Fig. 10A pile on in a specific outer-code.~~,~~ E~~e~~rror correction is impossible for the burst error in the inner-code error correction, and the number of errors exceeds the number of correctable outer-code. (Figs. 10B ~~and~~ to 10D will be described later.)

Moreover, in connection with the increase of high-density recording data for increasing capacity, one or two bytes of error due to dust and scratches of a current DVD become a short burst error. Thus, ~~a~~ the average length of time for error ~~concerned~~ correction (byte length) ~~becomes long~~increases, and it is estimated that the case that an uncorrectable error arises will increase.

Replace the paragraph beginning on page 7, line 17, with the following amended paragraph:

~~This~~ The invention disclosed in JP-A-8-125548 is a method of making ~~an~~ a burst error dispersed in the entire product code by rearranging the predetermined data in the product code every several bytes. Nevertheless, in the product code having a ratio of inner-code to outer-code like a DVD, there is a problem ~~not to be~~of not being able to secure correction capability, which is equivalent to a conventional method, for a~~n~~ burst error over ten or more lines. This is because there is a case that, since the burst error is dispersed in the outer-code direction by performing the rearrangement between lines, an area of uncorrectable errors ~~contrarily~~ increases. On the other hand, the present invention does not let an error become dispersed in an outer-code direction because the present invention does not rearrange the data between lines, and hence it is possible to maintain ~~correction capability~~the ability to correct for the burst error.

Replace

Thus, when the byte interleave for each row (here, the row ~~talks noting that it~~ has the same list as the data list recorded on the record medium) of a product code is performed under a rule different every row, that is, when data rearrangement is performed according to the rule which is different every inner-code of the ECC block 307 in a DVD, several bytes to several tens bytes of short burst error is dispersed in degree different every row. Hence, conventional errors shown in Fig. 10A become errors shown in Fig. 10B. Since an error count is equalized on outer-code in Fig. 10B even if uncorrectable errors are generated in the outer-code error correction in Fig. 10A, the probability of further decreasing the errors included in data by the outer-code error correction becomes high.

Replace the paragraph beginning on page 13, line 22, with the following amended paragraph:

Fig. 1 will be briefly described. The main ~~data recorded~~recorded data 310 is divided ~~per~~ into 2048 bytes. In each head, as shown in Fig. 4, 4 bytes of identification data (ID) 401, IED(ID error Detection Code) 402 that is ID error detecting code parity, and 6 bytes of RSV(reservation bytes) 403 that is a reservation area are added. Furthermore, 4 bytes of error detection code parity (EDC) is added to 2060 bytes of this data stream, then 2064 bytes of data unit (before scramble) 1 (304) is ~~configured~~organized into twelve rows of 172 bytes each. Next, according to a scramble rule determined with the value of the ID part of the data unit 1, scramble processing is performed ~~to~~ on the 2048 bytes of the main data part~~, and it becomes~~ to produce a data unit (after scramble) 1 (305) which is ~~12~~ twelve rows x 172 bytes. Furthermore, ~~16 bytes~~ sixteen rows of Parity of Outer-code (PO) data is added to a 192 row x 172 byte data field comprising ~~each of 172 columns (lengthwise) of 192 rows x 172 bytes that are configured by 16~~ sixteen data units 1 (306), to produce a 208-row data field ~~by making 16 blocks of these data units (after scramble) 1 (305) mutually combined~~. Next, ~~10~~ ten bytes of Parity of Inner-code (PI) is added to each of the 208 rows ~~(widthwise) containing PO~~ to form one ECC block (307) which is a ~~182 rows x 208 bytes~~ 208 row x 182 byte cross Reed-Solomon code. Next, in a PI interleave operation, the data in each row of the ECC block is rearranged ~~every one~~ on a byte-by-byte basis according to each rule to obtain an interleaved ECC block 101. Then, a PO interleave operation interleaves ~~which puts~~ each of the 16 rows of PO ~~parts (including 16 rows of~~and PI ~~for them)~~ data among the ~~into~~ sixteen data units 1 ~~as a 13th row is performed to form 16 data units 2 (102) (that is, one data unit 2 consists of 13 rows)~~ of the original 192 rows in the data field. Then, ~~after~~ 8/16 modulation ~~that~~ is performed to convert~~s~~ all data into 16 bits from 8 bits ~~is given~~, followed by the incorporation of twenty-six ~~26~~ SYNC codes ~~are added per one~~for each data unit 2 to form ~~16~~ data units 3 (103).

Replace the paragraph beginning on page 17, line 7, with the following amended paragraph

In Fig. 1, the added PI interleave operation is a ~~means~~ byte interleave ~~processing~~ operation based on a certain rule for rearranging the sequence ~~on the basis of a certain rule for~~ of 182 bytes of data (inner-code) of each row in the ECC block of DVD as shown in Fig. 2. That is, the added PI interleave operation shows the processing for rearranging the byte sequence about the data train which consists of a plurality of bytes constituting error correcting code. Here, the data train means an inner-code forming the ECC block, that is, an arrangement of the data in the row direction. At this time, the effect in the point of dispersion of errors becomes large by using conversion rules (interleave rules, $F_i(x)$) ~~Fi(x) (=y)~~ that have no correlation and are different from one another in the 208 rows of inner-code included in the ECC block. However, it is possible to obtain the effect of error dispersion even if this PI interleave is performed by using at least two kinds of conversion rules (interleave rules) ~~Fl (x)~~ $F_1(x)$ and $F_2(x)$ ~~F2 (x)~~. Moreover, data Di, 0; Di, 1; Di, 2; ...; Di, 181 show the data constituting the inner-code in the i-th row of an ECC block, and Di, 171; Di, 172; ...; Di, 181 correspond to the

parity of inner-code PI. Therefore, the data train (B) generated by performing PI interleave operation ~~to~~ on the data train (A) is not always inner-code.

Replace 5 paragraphs beginning on page 18, line 10, with the following amended paragraphs:

Fig. 11 is a diagram showing an ECC block (101) after performing the PI interleave shown in Fig. 1. This shows the status that each row of the ECC block shown in Fig. 6 is given PI interleave and is transformed.

Fig. 12 shows a data conversion flow at the time of recording data, generated after performing the data conversion process shown in Fig. 1, on a DVD and reproducing the data.

In Fig. 12, a data unit 2 (1203) is generated from the data that is recorded on the DVD while determining the position in the data unit 3 by using SYNC801, and performing 8/16 demodulation. Next, a PO deinterleave operation is performed to produce an ECC block (1201) ~~which is given PO interleave processing and is~~as shown in Fig. 11~~, is formed by canceling PO interleave~~ while determining the position of the data unit 2 in the ECC block 307 by using ID401. The PO deinterleave operation is the inverse of the operation shown in Fig. 1 where the PO row was inserted after every twelfth row. Then, the ECC block (1201) is restored to an ECC block (901) shown in Fig. 6 by applying the conversion rule $G_i(y)$ ~~Gi (y)~~, which is the inverse transformation of ~~Fi (x)~~$F_i(x)$ and is shown in Fig. 13. Thus, the bytes in each row are reorganized according to the inverse function $G_i(y)$ to restore ~~, that is, performing every row PI deinterleave which performs conversion of restoring again~~ the symbol train (A)~~, obtained by data rearrangement at the time of modulation, to PI code~~ from the symbol train (B). After this, as usual, an error of up to 5 bytes is corrected in each inner-code by the error correction to inner-code, and an error of up to 16 bytes, included in each outer-code, by the error correction to outer-code is corrected (306). Then, ID, IED, RSV, and 2048 bytes of main data (303) are obtained again after removing the data PI and PO for error correction required for error correction processing, restoring the ECC block to the 16 data unit 1 (305) shown in Fig. 5, canceling scramble (304), and performing error detection processing by using EDC. In addition, as for conversion rules $F_i(x)$~~Fi (x)~~ and $G_i(y)$ ~~Gi (x)~~, the apparatus may have the information as rules defined beforehand. Alternatively, the conversion rules $F_i(x)$~~Fi (x)~~ and $G_i(y)$~~Gi (x)~~ may be recorded ~~every~~ on the medium itself, and the apparatus may read the information. In the latter case ~~of the latter~~, it is possible to perform the above-described processing after reading conversion rules $F_i(x)$~~Fi (x)~~ and $G_i(y)$~~Gi (x)~~ first, and to determine the conversion rules $F_i(x)$~~Fi (x)~~ and $G_i(y)$~~Gi (x)~~ for every disk unit, ECC block unit, frame unit, or the like in the apparatus.

However, in the demodulation process shown in Fig. 12, the position in the ECC block is determined by using the ID. Thus, if a PI interleave operation is performed where the interleave function $F_i(x)$ varies for each row, it is not possible to determine $x=G_i(y)$ at the time of restoring a data train since a conversion rule $y=F_i(x)$ for PI interleave for a row including ID cannot be found. Hence, it becomes very difficult to

seek the ID.~~However, as shown in Fig. 12, in case demodulation processing is usually performed in DVD used as an example in this explanation, the position in the ECC block is determined by using ID. Hence, when PI interleave which is different in every row of all ECCs is applied, if PI interleave processing at the time of record is applied, it is not possible to determine x=Gi(y) used in PI deinterleave at the time of restoring a data train since a conversion rule y=Fi (x) for PI interleave for a row including ID cannot be found. Hence, it becomes very difficult to seek the ID.~~

Therefore, the information showing the position in a product code is shown, that is, ID and IED (IED is also required when the reliability of ID is required) are ~~removed from the objects of~~not subjected to any data interleave operation on a DVD. Consequently, ~~and~~ rows including this information are ~~given~~ subject PI interleave processing by processing only ~~to~~ 176 bytes (e.g., bytes 7 through 182) of data ~~except~~ and leave the 6 bytes of identification information such as ID and IED unaltered.

Replace 5 paragraphs beginning on page 21, line 1, with the following amended paragraphs:

Moreover, as shown in Fig. 14, if removing ID and IED from the objects of PI interleave, in regard to the conversion rules $F_i(x)$~~Fi (x)~~ of the inner-code including ID and IED, it is assumed that x= 1, ..., 182 and the values of x show position addresses as shown in Fig. 2. The~~n it should~~ function can be ~~devised~~ revised to ~~become~~ so that ~~Fi~~$F_i(n)$ =n (for n= 1, 2, 3, 4, 5, and 6), or ~~it is~~ the function can made to limit x ~~and Fi (x)~~ to be in ~~a~~ the range of 6 to 182.

Fig. 2 on which is briefly touched above will be described again. Fig. 2 is an example of a diagram having shown the method of a data list at the time of adopting PI interleave with the conversion rule $F_i(x)$~~Fi (x)~~ for PI interleave to the inner-code described in Fig. 6 where the present invention is described above. An inner-code is 182 bytes of data train, data (A) denotes the inner-code after the addition of PI and PO, and data (B) denotes 182 bytes of data train which is given PI interleave by the conversion rule $F_i(x)$~~Fi (x)~~ and is rearranged in the sequence of data thereof. Here, although the data (B) is an example of rearranging data every 13 pieces, the conversion rule $F_i(x)$~~Fi (x)~~ is not limited to this, but is enough to discontinuously rearrange the data.

Two or more kinds of conversion rules $F_i(x)$~~Fi (x)~~ are prepared in order to perform different values of PI interleave. In a DVD, the conversion rules of up to 208 kinds are prepared, and error counts on outer-codes in an ECC block are equalized as shown in Figs. 10A to 10D by applying the different conversion rule for every inner-code. Hence, cases that error correction to outer-codes become possible increases. Thus, it becomes possible to increase correction capacity by performing different PI interleave to the inner-codes and recording the inner-codes on a record medium. Moreover, this effect becomes still larger when the code is decoded repeatedly.

Next, a method of implementing the conversion rule for this PI interleave in a circuit will be described. First, an example of a configuration of a DVD recording and reproducing apparatus ~~at the time of applying~~according to the present invention will be described by using Fig. 15. Although a recording and reproducing apparatus is described

as an example here, the~~a~~ present invention is applicable also to a reproduction dedicated device and a record dedicated device.

An optical pickup 1502 performs record and reproduction of data on a record medium 1501 such as a DVD, and a spindle motor 1503 rotates a disk. Moreover, a servo 1504 controls the optical pickup 1502 or the like. A read channel 1505 performs waveform equalization and doubling of an analog reproductive signal read from the record medium 1501, and synchronous clock generation. A decoder 1506 consists of a demodulation circuit 1507 performing 8/16 demodulation of data read, and an error correcting circuit (1508, 1518) performing the processing of removing errors included in the data, and RAM 1509a temporarily stores the data at the time of reproduction. RAM 1509b temporarily stores the data at the time of record. This can be also used as the RAM 1509a. Numeral 1514 is a laser driver. A modulation circuit 1512 performs the modulation of the data at the time of record, and an encoding circuit 1513 consists of an error correcting code generation circuit adding parity of error correcting code, PI and PO, and a scramble circuit performing scramble processing. An interface 1515 controls the input/output of data with a host system, and a microcomputer 1516 generalizes the system. In addition, it is also possible to use a microcomputer as the error correcting code generation circuit adding the error correcting code parity PI and PO.

Replace 2 paragraphs beginning on page 25, line 9, with the following amended paragraphs:

Next, a PI interleave circuit and a PI deinterleave circuit that are shown in Fig. 15 will be described. Fig. 19 shows an example of the PI interleave circuit 1517 where PI interleave conversion rules are defined by using the M-series generating circuit 1901~~(~~, which is a circuit ~~generating~~ that generates the maximum periodic column~~, and~~; e.g., the maximum period column ([=]referred to as the M-series) is 255 ~~at the time~~in the case of using an 8-bit register (i.e., 2^8-1). Moreover, here, it is assumed that the inner-code read from the RAM 1509b has been already stored in a shift register 1902. As write control signals to the RAM 1509b, there are a write request signal, an address (however, referred to as 1 to 182 for convenience here) in the RAM 1509b, and write data. When the write request signal is outputted, a write request acknowledge signal, which shows that data is written, is inputted from the RAM 1509b.

In this circuit, when the write request acknowledge signal is inputted, that is, the timing when the data is written, the shift register 1902 shifts. Moreover, the M-series generating circuit 1901 also changes to the following value if the write request acknowledge signal is inputted, or if the signal of 1 to 255 generated by the M-series generating circuit 1901 exceeds the address to be required, i.e., if the signal has the value of 183 or more. Here, the value range is not necessary to be 1 to 255. In a DVD, the inner-code is 182 bytes, and PI interleave in the data in all the inner-codes is concerned here. Hence, an output from the 8-bit M-series generating circuit is used. Moreover, the write request signal is outputted when the address made in the M-series generating circuit 1901 is ~~judged~~ determined to be 182 or less by a request generation judging circuit 1903.

Replace the paragraph beginning on page 27, line 17, with the following amended paragraph:

In this circuit also, it is necessary to ~~make~~ skip an address ~~skipped~~ if the address made in the M-series generating circuit 1901 exceeds 183, because PI deinterleave processing cannot be performed.

Replace the paragraph beginning on page 28, line 28, with the following amended paragraph

Moreover, different M-series can be also generated by changing the ~~polarity~~ value of a certain bit as shown in Fig. 17B. However, since zero may be outputted from M-series in this case, and a value at the time of the register being zero and an address is not generated at the value "AA=170" in hexadecimal notation in Fig. 17B, it becomes necessary to ~~perform such a devise that an address is calculated by subtracting~~ 0x170 from the obtained value to produce a suitable address. However, these can be used as M-series that define PI interleave rules. Moreover, similarly, it is clear that an address can be converted by using a logic circuit obtained by combining OR, AND, and NOT.

Replace the paragraph beginning on page 35, line 18, with the following amended paragraph

In addition, it should be noted that, when recording to a record media is performed according to the list of the outer-code in a product code, the inner-code used in the explanation corresponds to the outer-code. Moreover, although the signal processing circuit using an M-series and an arithmetic progression is exemplified in the PI interleave circuit 1517 and PI deinterleave circuit 1518, other pseudo-random number generating circuits or the like can be used.